CSN-341
Computer Networks
Project Report

Group-6

Name	Enrollment No
Meet Sindhav	22114053
Mohammed Haaziq	22114055
Aditya Mundada	22114058
Nayan Kakade	22114060
Sarvesh Baheti	22114087
Roopam Taneja	22125030

SNMP Network Management Tool

Overview

The SNMP tool developed by our team is capable of:

- Monitoring client metrics such as CPU usage, memory usage, Wi-Fi incoming packets, and more.
- Configure certain client metrics like client name, location, etc.
- Enabling the client to send traps (real-time alerts) to the server about critical events like system reboots or high CPU load.
- Visualizing the data received from the client by plotting graphs, correlating time with the respective metrics.
- Log the received traps into a text file

Theory & Background

Why is Network Management Needed?

Network management plays a critical role in ensuring the reliability, security, and efficiency of modern networks. The following points illustrate why network management is essential, based on real-world scenarios:

- **Fault Detection and Troubleshooting:** Network management allows for quick identification of issues, whether they are related to hardware failures, services being unresponsive, or misbehaving interfaces.

  Effective troubleshooting minimizes user impact, ensuring the network stays functional and performs optimally.
- **Capacity Planning:** As network traffic grows, the ability to monitor and forecast bandwidth requirements is crucial. Network management provides the data needed to anticipate increases in traffic, plan upgrades in advance, and avoid congestion that could otherwise disrupt operations.

- **Security Compliance:** Maintaining security standards is essential for any organization. Network management helps ensure compliance with regulations (such as PCI-DSS) by identifying devices or areas of the network that may be out of compliance.

  This includes detecting non-SSL traffic or unencrypted protocols, which can present security vulnerabilities.

- **Performance Optimization:** Users expect fast response times and smooth network operation. By monitoring network performance, it's possible to detect bottlenecks and optimize traffic flows. For example, protocol breakdowns can help create Quality of Service (QoS) policies to prioritize critical traffic.

- **Support and Upgrades:** Keeping the network up to date is a continual process. Network management tools help track devices and software versions, ensuring that outdated or vulnerable equipment is replaced or patched.

  This is crucial for maintaining security and taking advantage of new features or improvements.

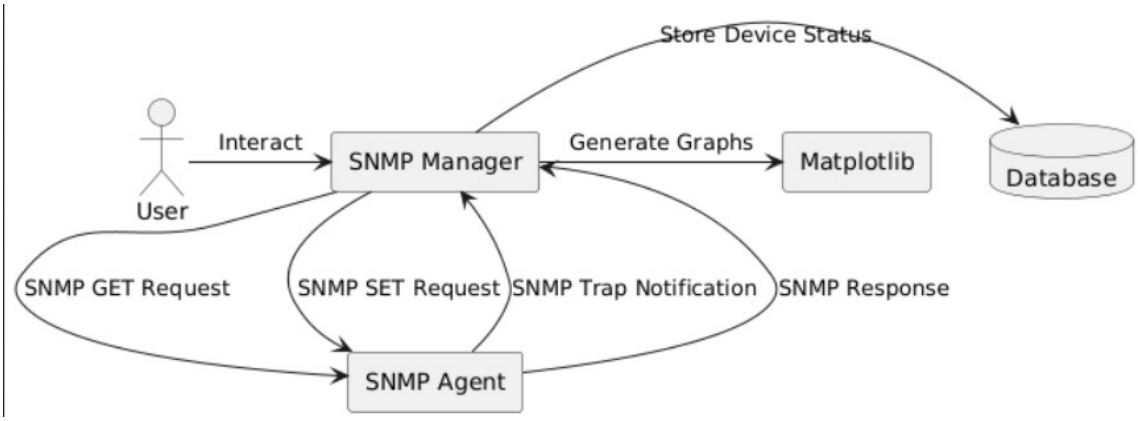
What is SNMP?

Simple Network Management Protocol (SNMP) is a standard protocol used to collect and organize information about managed devices on IP networks. It allows network administrators to monitor and manage network performance, detect network faults, and configure remote devices.

It is an application-level protocol in which a few manager stations control a set of agents. The protocol is designed at the application level so that it can monitor devices made by different manufacturers and installed on different physical networks.

In other words, SNMP frees management tasks from both the physical characteristics of the managed devices and the underlying networking technology.

Our approach relies on the APIs provided by the Python `pysnmp` library such as `snmpget`, `snmpset`, and `snmptrap`. We have configured our system's SNMP files to allow sending and receiving of SNMP commands.



Details of Implementation

Project Directory Structure

```
SNMP Network Management Tool
├── Readme
├── graph.py
├── info.csv
├── snmpManager.py
├── trap_client_demo.py
├── trap_receiver.py
└── trap_sender.py
```

Functions performed by each file

- **graph.py:** Graphically represents the collected data statistics from each device. This program collects data from a .csv file corresponding to each device being monitored and depicts it in a visually appealing manner.
  Libraries: pandas, matplotlib
- **info.csv:** Information collected from each device will be stored in a csv file, which will be used by graph.py for plotting
- **trap_client_demo.py:** This Python script acts as the SNMP trap client. It is responsible for sending trap messages to the SNMP server whenever a specific event or condition occurs. This client can be configured to generate traps based on certain thresholds or incidents detected in the monitored environment
  Libraries: asyncio, pysnmp
- **trap_receiver.py:** This Python script functions as the SNMP trap receiver. It listens for incoming trap messages from clients and processes them accordingly. The receiver can log the received traps, trigger alerts, or perform other actions based on the information contained in the traps.
  Libraries: pysnmp, logging
- **snmpManager.py:** The SNMP Manager File includes a function called **get_device_info** that retrieves system information from an SNMP-enabled device using its IP address. The function uses a set of Object Identifiers (OIDs) to gather metrics like system description, location, packet statistics, CPU utilization, and memory usage. It efficiently handles communication errors and displays the retrieved information in a clear and understandable format, making it easy to monitor and manage network devices. The function uses **asynchronous** execution, which allows it to perform other tasks while retrieving information, improving performance and responsiveness.
  Libraries: asyncio, logging, pysnmp, csv, sys
- **trap_sender.py:** This Python script sets a threshold for memory usage, above which a trap will be sent from client to the manager to report excessive memory usage or a temperature rise. This limit is customisable, for eg, 50% for memory usage and 60 °C for temperature.
  Libraries: asyncio, psutil

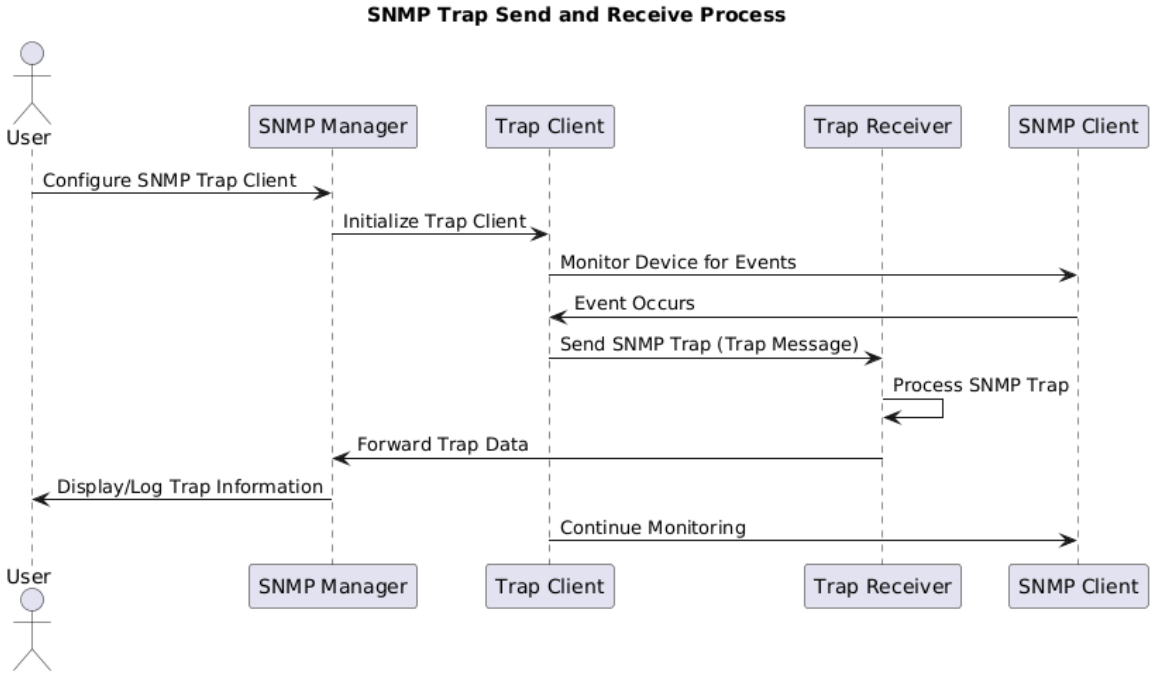
Source Code: Link

Limitations

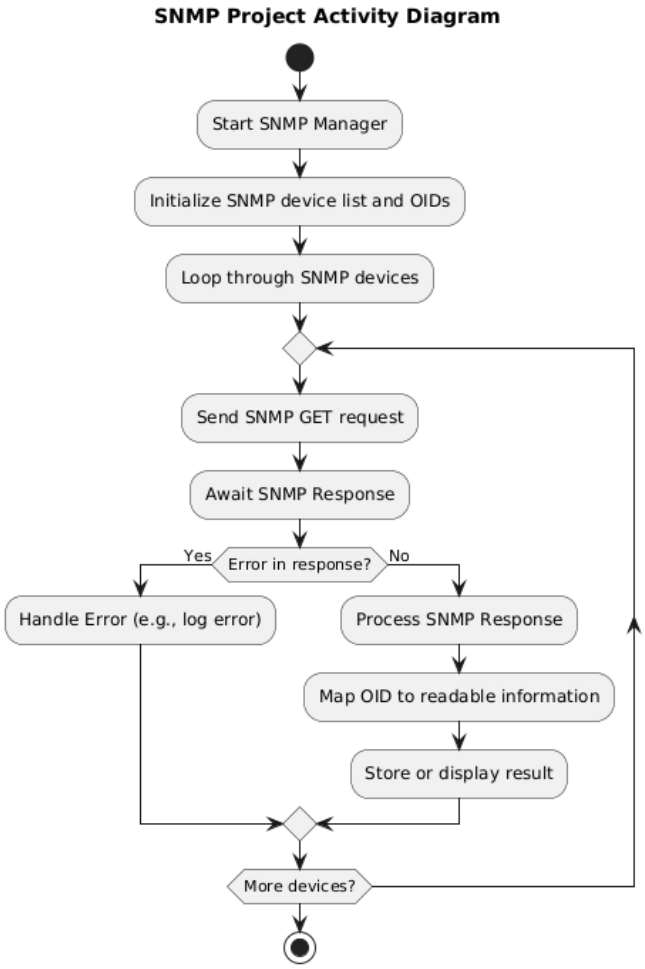
- Has not been implemented routers and switches due to accessibility constraints.
- Not possible to test traps for specific hardware failures, such as when the NIC (Network Interface Card) malfunctions.
- Router configuration is required to scale the project to devices connected across different networks.
- Lacks encryption as protocol used is SNMP v2c, not v3.
- Vulnerable to attacks like DoS (Denial of Service), where a few clients can overload the server by sending too many traps.

Images and Diagrams

Sequence Diagram: SNMP Trap Management



Activity Diagram: SNMP Manager



SNMP Traps

```
CPU usage: 5.5%
Memory usage: 69.1%
Memory threshold exceeded: 69.1%, sending SNMP trap...
Trap sent successfully! Memory usage exceeded!
CPU usage: 12.1%
CPU threshold exceeded: 12.1%, sending SNMP trap...
Trap sent successfully! CPU usage exceeded!
Memory usage: 73.4%
Memory threshold exceeded: 73.4%, sending SNMP trap...
Trap sent successfully! Memory usage exceeded!
CPU usage: 7.3%
Memory usage: 71.9%
Memory threshold exceeded: 71.9%, sending SNMP trap...
Trap sent successfully! Memory usage exceeded!
CPU usage: 6.3%
Memory usage: 72.3%
Memory threshold exceeded: 72.3%, sending SNMP trap...
Trap sent successfully! Memory usage exceeded!
CPU usage: 6.0%
Memory usage: 71.6%
Memory threshold exceeded: 71.6%, sending SNMP trap...
Trap sent successfully! Memory usage exceeded!
CPU usage: 3.4%
Memory usage: 71.1%
```

Figure 1: Logging information for trap sender

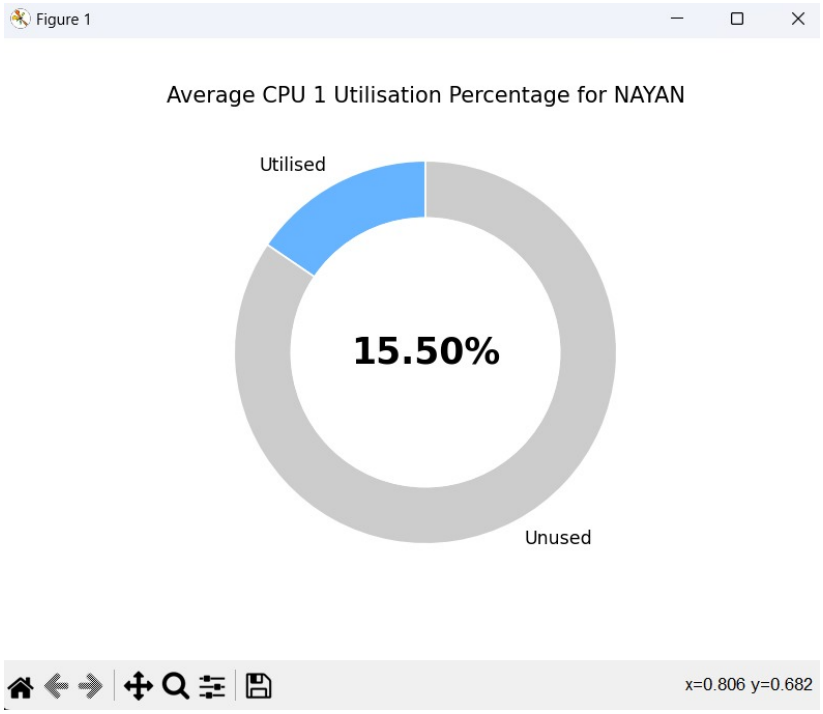
outputlog.txt

```
00:43:58.852725 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(136)  system.sysUpTime.0=40866 S:1.1.4.1.0=system.sysDescr.0 system.
00:43:59.823904 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(113)  system.sysUpTime.0="CPU usage exceeded: 12.8%" S:1.1.4.1.0=sy
00:43:59.853908 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(142)  system.sysUpTime.0=40969 S:1.1.4.1.0=system.sysObjectID.0 syst

00:44:06.119513 wlp4s0 In  IP 10.81.55.159.49671 > nayan-Inspiron-3501.snmp-trap:  V2Trap(137)  system.sysUpTime.0=33037 S:1.1.4.1.0=system.sysDescr.0 syst
00:44:09.908774 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(136)  system.sysUpTime.0=41970 S:1.1.4.1.0=system.sysDescr.0 system.
00:44:10.934978 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(142)  system.sysUpTime.0=42073 S:1.1.4.1.0=system.sysObjectID.0 syst
00:44:17.076538 wlp4s0 In  IP 10.81.55.159.49671 > nayan-Inspiron-3501.snmp-trap:  V2Trap(137)  system.sysUpTime.0=34138 S:1.1.4.1.0=system.sysDescr.0 syst
00:44:20.967950 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(136)  system.sysUpTime.0=43073 S:1.1.4.1.0=system.sysDescr.0 system.
00:44:21.992325 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(142)  system.sysUpTime.0=43176 S:1.1.4.1.0=system.sysObjectID.0 syst
00:44:28.135777 wlp4s0 In  IP 10.81.55.159.49671 > nayan-Inspiron-3501.snmp-trap:  V2Trap(137)  system.sysUpTime.0=35240 S:1.1.4.1.0=system.sysDescr.0 syst
00:44:32.027228 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(136)  system.sysUpTime.0=44177 S:1.1.4.1.0=system.sysDescr.0 system.
00:44:33.051172 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(142)  system.sysUpTime.0=44279 S:1.1.4.1.0=system.sysObjectID.0 syst
00:44:39.092785 wlp4s0 In  IP 10.81.55.159.49671 > nayan-Inspiron-3501.snmp-trap:  V2Trap(137)  system.sysUpTime.0=36341 S:1.1.4.1.0=system.sysDescr.0 syst
00:44:42.983935 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(136)  system.sysUpTime.0=45280 S:1.1.4.1.0=system.sysDescr.0 system.
00:44:44.111700 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(142)  system.sysUpTime.0=45385 S:1.1.4.1.0=system.sysObjectID.0 syst

00:44:50.152907 wlp4s0 In  IP 10.81.55.159.49671 > nayan-Inspiron-3501.snmp-trap:  V2Trap(137)  system.sysUpTime.0=37442 S:1.1.4.1.0=system.sysDescr.0 syst
00:44:54.043531 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(136)  system.sysUpTime.0=46385 S:1.1.4.1.0=system.sysDescr.0 system.
00:44:55.067452 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(142)  system.sysUpTime.0=46489 S:1.1.4.1.0=system.sysObjectID.0 syst
00:45:01.118149 wlp4s0 In  IP 10.81.55.159.49671 > nayan-Inspiron-3501.snmp-trap:  V2Trap(137)  system.sysUpTime.0=38542 S:1.1.4.1.0=system.sysDescr.0 syst
00:45:05.123401 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(136)  system.sysUpTime.0=47489 S:1.1.4.1.0=system.sysDescr.0 system.
00:45:06.054131 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(113)  system.sysUpTime.0="CPU usage exceeded: 10.7%" S:1.1.4.1.0=sy
00:45:06.078021 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(142)  system.sysUpTime.0=47592 S:1.1.4.1.0=system.sysObjectID.0 syst
00:45:16.162170 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(136)  system.sysUpTime.0=48592 S:1.1.4.1.0=system.sysDescr.0 system.
00:45:17.186111 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(142)  system.sysUpTime.0=48696 S:1.1.4.1.0=system.sysObjectID.0 syst
00:45:27.221543 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(136)  system.sysUpTime.0=49696 S:1.1.4.1.0=system.sysDescr.0 system.
00:45:28.246193 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(142)  system.sysUpTime.0=49799 S:1.1.4.1.0=system.sysObjectID.0 syst
00:45:38.178442 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(136)  system.sysUpTime.0=50800 S:1.1.4.1.0=system.sysDescr.0 system.
00:45:39.204689 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(142)  system.sysUpTime.0=50903 S:1.1.4.1.0=system.sysObjectID.0 syst
00:45:49.237897 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(136)  system.sysUpTime.0=51903 S:1.1.4.1.0=system.sysDescr.0 system.
00:45:50.261730 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(142)  system.sysUpTime.0=52006 S:1.1.4.1.0=system.sysObjectID.0 syst
00:46:00.296959 wlp4s0 In  IP 10.81.72.3.45923 > nayan-Inspiron-3501.snmp-trap:  V2Trap(136)  system.sysUpTime.0=53007 S:1.1.4.1.0=system.sysDescr.0 sys
```

Figure 2: An instance of received traps

CSV Data Extracted

Timestam	System na	System De	System Lo	Wifi incom	Ethernet ir	Wifi outgo	Ethernet o	Wifi inbou	Ethernet ir	Wifi outbo	Ethernet o	System Up	Number of	CPU 1 Util	CPU 2 Util	Total Mem	Memory used
########	ADITYA	Linux Adity	Sitting on t	67956241	0	10405160	0	0	0	0	0	1577496	378	1	1	15592276	6702580
########	ADITYA	Linux Adity	Sitting on t	67956778	0	10405862	0	0	0	0	0	1578027	378	1	1	15592276	6699288
########	ADITYA	Linux Adity	Sitting on t	67957394	0	10406577	0	0	0	0	0	1578574	378	1	1	15592276	6690736
########	ADITYA	Linux Adity	Sitting on t	67958042	0	10407454	0	0	0	0	0	1579087	378	1	1	15592276	6667296
########	ADITYA	Linux Adity	Sitting on t	67958487	0	10408064	0	0	0	0	0	1579604	378	1	1	15592276	6652068
########	ADITYA	Linux Adity	Sitting on t	67958932	0	10408674	0	0	0	0	0	1580126	378	1	1	15592276	6652312
########	ADITYA	Linux Adity	Sitting on t	67959377	0	10409284	0	0	0	0	0	1580664	375	1	1	15592276	6651628
########	ADITYA	Linux Adity	Sitting on t	67960091	0	10410161	0	0	0	0	0	1581191	375	1	1	15592276	6651888
########	ADITYA	Linux Adity	Sitting on t	67961592	0	10411263	0	0	0	0	0	1581715	375	1	1	15592276	6728792



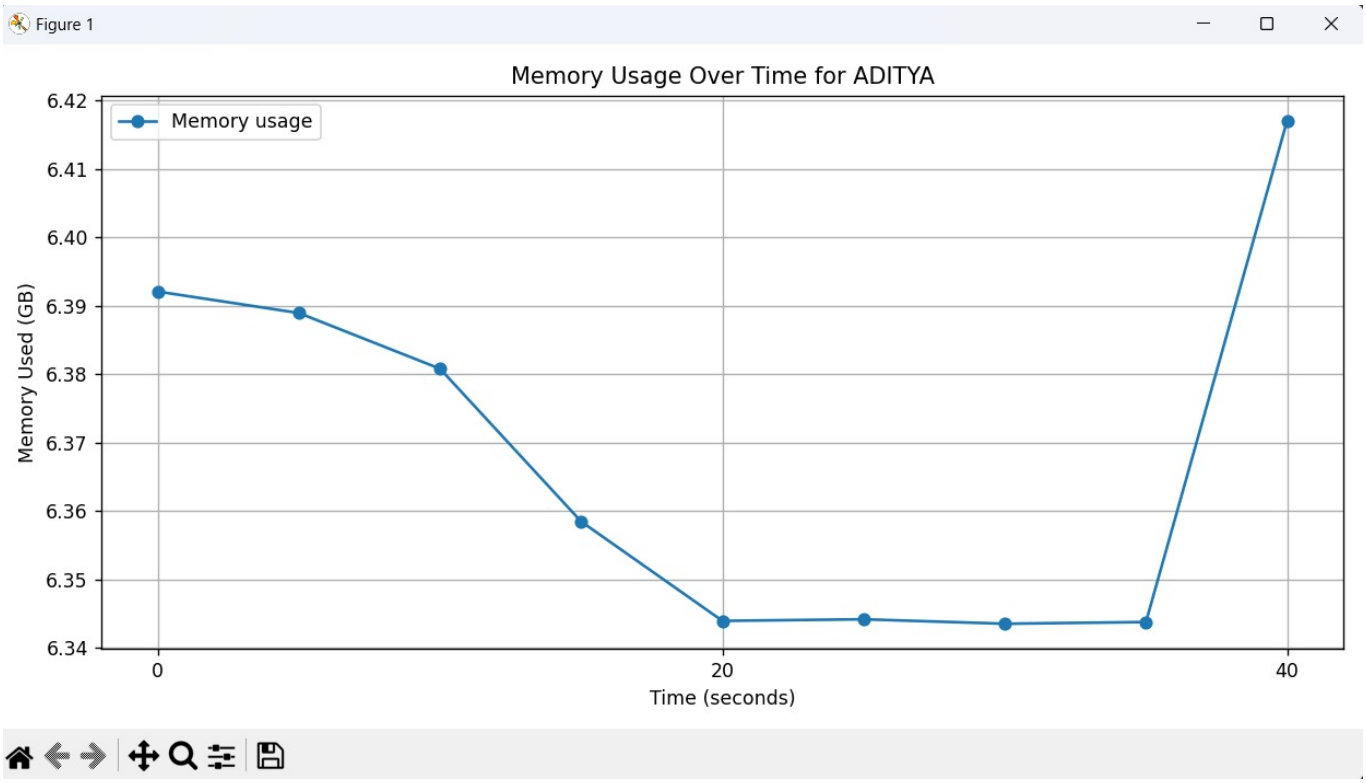


Figure 1
Memory Usage Over Time for ADITYA
Memory usage
Memory Used (GB)
6.42
6.41
6.40
6.39
6.38
6.37
6.36
6.35
6.34
0
20
40
Time (seconds)

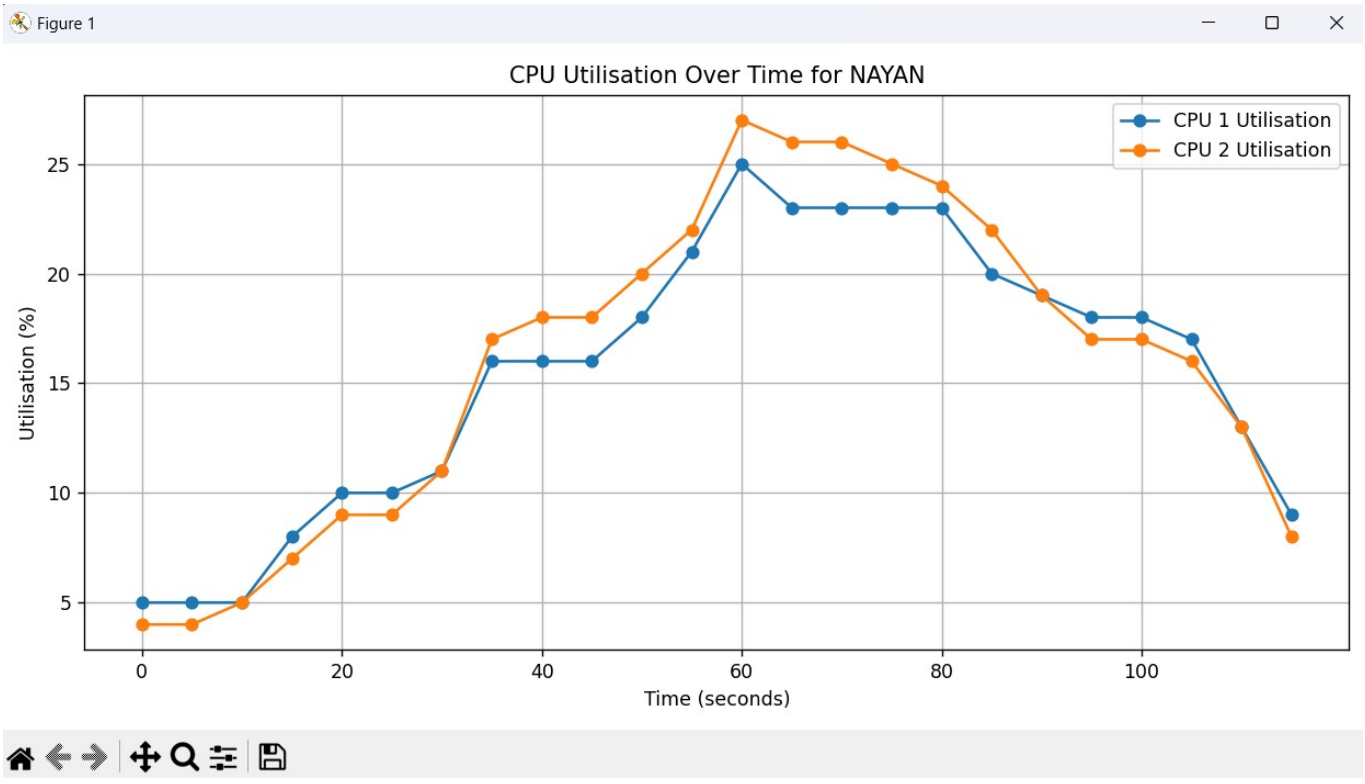


Figure 1
CPU Utilisation Over Time for NAYAN
CPU 1 Utilisation
CPU 2 Utilisation
Utilisation (%)
25
20
15
10
5
0
20
40
60
80
100
Time (seconds)